NPS-CS-99-005

# NAVAL POSTGRADUATE SCHOOL
# Monterey, California

**An Approach to Characterizing Resource Usage and User Preferences in Benefit Functions**

by

Timothy Levin
Cynthia E. Irvine

15 June 1999

Approved for public release; distribution is unlimited.

Prepared for: Defense Advanced Research Project Agency
and Information Technology Organization

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

RADM Robert C. Chaplin
Superintendent

R. Elster
Provost

This report was prepared as part of the Naval Postgraduate School Center for Information Systems Security (INFOSEC) Studies and Research (NPS CISR) at the Naval Postgraduate School, as part of a project funded under the Defense Advanced Research Projects Agency/Information Technology Organization grant under the Quorum program.

Cynthia E. Irvine

Cynthia Irvine
Assistant Professor
Department of Computer Science

Tim Levin
Senior Research Associate

Reviewed by:

Neil Rowe
Associate Professor
Department of Computer Science

Released by:

Dean D. Boger, Chair
Department of Computer Science

D. W. Netzer
Dean of Research

| REPORT DOCUMENTATION PAGE | | | Form approved<br>OMB No 0704-0188 |
|---|---|---|---|



| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE<br>15 June 1999 | 3. REPORT TYPE AND DATES COVERED<br>Progress; 4/15/99 – 6/15/99 |
|---|---|---|

| | |
|---|---|
| 4. TITLE AND SUBTITLE<br>An Approach to Characterizing Resource Usage and User Preferences in Benefit Functions | 5. FUNDING<br>MIPR 99-E583 |
| 6. AUTHOR(S)<br>TIMOTHY LEVIN AND CYNTHIA IRVINE | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br>Naval Postgraduate School, 833 Dyer Road, Monterey, CA 93943 | 8. PERFORMING ORGANIZATION REPORT NUMBER<br>NPS-CS-99-005 |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)<br>DARPA/ITO<br>3701 North Fairfax Drive<br>Arlington, VA 22203-1714 | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER |
| 11. SUPPLEMENTARY NOTES | |
| 12a. DISTRIBUTION/AVAILABILITY STATEMENT<br>Approved for public release; distribution is unlimited | 12b. DISTRIBUTION CODE |

13. ABSTRACT (Maximum 200 words.)

An approach is described for representing the level of resources consumed by jobs under the control of a Resource Management System, and it is shown how this measurement of resource usage can be combined with a notion of user preferences to reflect a restrictive resource-usage policy for network management.

| | |
|---|---|
| 14. SUBJECT TERMS<br>resource management system, benefit functions, resource usage, engineering and assessment, metric | 15. NUMBER OF PAGES 11 |
| | 16. PRICE CODE |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassifed | Unclassified | Unclassified | Unlimited |

NSN 7540-01-280-5800

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18

# An Approach to Characterizing Resource Usage and User Preferences in Benefit Functions[1]

**Tim Levin**
**Anteon Corporation**
**Monterey, CA**

**Cynthia Irvine**
**Naval Postgraduate School**
**Monterey, CA**

**Abstract.** *An approach is described for representing the level of resources consumed by jobs under the control of a Resource Management System, and it is shown how this measurement of resource usage can be combined with a notion of user preferences to reflect a restrictive resource-usage policy for network management.*

## 1 Introduction

Various mechanisms exist for managing contention and allotment of distributed network resources. One class of these mechanisms attempts to schedule, in the most efficient means possible, the execution of multiple, simultaneous, jobs on multiple distributed, heterogeneous, computers [1] [2] [3] [9] [10] , where each job requires a determinable subset of the resources.

Abstract benefit functions can be used to measure the effectiveness of resource management systems in satisfying various system and user requirements in the operation of a virtual heterogeneous network. Such measurements can be used to drive the RMS scheduling mechanism, as well as to study the behavior of the RMS.

Of interest is a function for comparing the relative benefits of job scheduling mechanisms when they are presented with real or hypothetical "data sets" of jobs. As we are not considering a "benchmark" type of definition which has a predefined data set, the benefit function needs to be fair regarding the nature of the resource consumption attempted. For example, when comparing mechanism "A," scheduling a set of jobs which require 50% of the resources, to mechanism "B," scheduling a set of jobs which require 98% of the resources, the benefit function needs to give more credit for scheduling the more difficult data set (B, in this example).

We develop, in Section 2, an efficiency metric for showing the effectiveness of an RMS in scheduling jobs with respect to resource usage. Usage is represented as a ratio of *resources scheduled* to *resources available*. Then, in Section 3 a benefit function is developed, utilizing the efficiency metrics as well as representations of a job's priority and preference. A conclusion follows in section 4.

## 2 Resource Usage Efficiency Metric

In the network computing context, users or user programs may request the execution of "jobs," which are scheduled by an underlying control program to execute on local or remote computing resources. The execution of the job may access or consume a variety of network resources, such as: local I/O device bandwidth; internetwork bandwidth; local and remote CPU time; local, inter-

1. Funded through MSHN, a DARPA/QUORUM project.

mediate (e.g., routing buffers) and remote storage. The resource usages may be temporary or may persistent for the duration of the job. As there are multiple users accessing the same resources, there are naturally various allotment, contention, and security issues associated with the use of those resources.

The jobs in a particular "data set" are represented by the set $J$. The number of jobs in $J$ is $n$. To characterize resource usage, we will abstract both time and resource usage. We measure time in *Time Units.* The total amount of a given resource available during a time unit is one *Resource Unit.* Note that this abstract unit will be used to measure the proportion of a resource consumed by a job, rather than the magnitude of a resource in a given time unit. Each job has an associated deadline, before which it must finish. Deadlines are measured in time units, from the data set's start time. Thus, some jobs may be delayed to start later than other jobs, but this does not affect the deadlines of the delayed jobs. The length of the longest deadline in $J$, is $T$. This can be understood as the overall deadline for the data set. The number of different resources available is $[R]$. The total number of resource units available over $T$ is:

$$T \times [R]$$

During each Time Unit ($0 <= t <= T$), a job ($j$) requires a fractional amount ($0 <= C_{jrt} <= 1$) of each available resource ($r$). A resource is considered applicable to a job if and only if the job requires some but not more than 100% of the resource. The relationship of time, resources and jobs can be represented in a three-dimensional matrix, as in Table 1 . Here, we see that in time

**Table 1: Example Resources Required per job and time unit**

| Resources | Jobs | Time Units (T = 5) | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Resource 1 | job 1 | .5 | .8 | 0 | .5 | .5 |
| | job 2 | .5 | .5 | .8 | .2 | 0 |
| Resource 2 | job 1 | .2 | .2 | .5 | .2 | .2 |
| | job 2 | .3 | .5 | .7 | .8 | 0 |

unit 2, job 1 requires 80% of resource 1 and 20% of resource 2. Notice, that there are some resource conflicts between these two jobs. In time unit 2 for resource 1 and time unit 3 for resource 2, the two jobs require more than 100% of the available resources. The RMS could resolve this conflict, for example, by delaying the start of job 2 for one time unit.

The fraction of resources "required" to resources "available" over the whole data set is:

$$K_{ideal} = \sum_{t=1}^{T} \sum_{j=1}^{n} \sum_{r=1}^{[R]} C_{jrt}/(T[R])$$

If the numerator of the above expression is greater than the denominator, then the jobs cannot be scheduled within the specified deadlines; otherwise, $0 <= K_{ideal} <= 1$.

Now, a job will succeed in scheduling only a certain fraction ($K_{actual}$) of the resources that it

requires. For example, a job may run in a degraded format. We introduce the variable Q to indicate the actual scheduling of a given resource:

$\forall t \leq T,\ r \in R,\ j \in J$ (if resource *r* is scheduled for job *j* in time *t*, then $Q_{jrt} = C_{jrt}$, else $Q_{jrt} = 0$)

The model is that a job either gets all of the resources it requires ($C_{jrt}$) in a time unit, or none of it. Degradation with respect to required resources may occur over time, but not within a time unit.

$K_{actual}$ is the ratio of resources scheduled to resources required by all jobs:

$$K_{actual} = \sum_{t=1}^{T} \sum_{j=1}^{n} \sum_{r=1}^{[R]} Q_{jru} / C_{jru}$$

$0 <= K_{actual} <= 1$

As stated, the jobs the scheduler attempts to schedule require some fraction ($K_{ideal}$) of the number of available resources. Intuitively, $K_{actual}$ and $K_{ideal}$ look like this:

$K_{actual}$ = number of scheduled resource units / number of required resource units
$K_{ideal}$ = number of required resource units / number of available resource units

Recall that we wanted to temper the measurement of a mechanism's success at scheduling (viz, $K_{actual}$) with a notion of how hard of a job it had attempted (viz, $K_{ideal}$). The efficiency of a network job-scheduling mechanism can be characterized by multiplying the success rate by the difficulty rate:

$$\begin{aligned} \mathit{Efficiency} &= K_{actual} \times K_{ideal} \\ &= \text{number of scheduled resource units / number of available resource units} \\ &= \sum_{t=1}^{T} \sum_{j=1}^{n} \sum_{r=1}^{[R]} Q_{jrt} / (T[R]) \end{aligned}$$

$0 <= \mathit{Efficiency} <= 1$

For example, the set of jobs to be scheduled by a mechanism require 80% of the available resources. It succeeds in scheduling 90% of its required resources. The *Efficiency* of the mechanisms is:

.9 x .8 = .72

## 2.1 Job-Scheduling Examples

We will illustrate this notion of efficiency with two simplified network job-scheduling mechanisms. The first mechanism (mechanism #1) knows how to utilize multiple CPUs. The second mechanism only knows how to schedule jobs sequentially on one CPU. There are two data sets to be measured against each mechanism. Since mechanism 1 is smarter, we expect it to be more

efficient than mechanism 2.

**Table 2: Data Set #1**

| | CPU | % of Memory | % of Total Bandwidth | Deadline (time units) | Resource Units Required |
|---|---|---|---|---|---|
| Job 1 | 8 units | 50 (4 units) | 50 (4 units) | 10 | 16 |
| Job 2 | 5 units | 50 (2.5 units) | 50 (2.5 units) | 8 | 10 |
| Total | 13 units | 6.5 units | 6.5 units | -- | 26 |

Data set one has two jobs. The first job requires 8 time units of CPU usage (this could be qualified for different CPU speeds), meaning that the job can finish in 8 time units if it has full access to a CPU. This equates to 8 resource units. It requires 50% of the available memory while it is executing, equating to 4 resource units of memory. It requires 50% of the network bandwidth while it is executing, again equating to 4 resource units of bandwidth. Job 1 requires completion in 10 time units after starting. Job two is similar, except that it requires less CPU time, and has a shorter deadline. The two jobs require a total of 26 resource units. The length of the longest deadline (T) is 10. For resources, there are two CPUs, memory and network bandwidth, each 100% available for the duration of T, yielding 40 available resource units[1].

The second data set is the same as the first, except that the deadline for job 2 is increased to 14 time units. There are 52 available resource units.

**Table 3: Data Set #2**

| | CPU | % of Memory | % of Total Bandwidth | Deadline | Resource Units Required |
|---|---|---|---|---|---|
| Job 1 | 8 units | 50% (4 units) | 50% (4 units) | 10 | 16 |
| Job 2 | 5 units | 50% (2.5 units) | 50% (2.5 units) | 14 | 10 |
| Total | 13 units | 6.5 units | 6.5 units | -- | 26 |

Table 4 shows the result of submitting data set 1 to mechanism 1. The efficiency of this mechanism/data set is:

$$Efficiency = 26/40 = .65$$

**Table 4:** Mechanism 1, Data Set 1, T = 10, 40 available units

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CPU 1 | | | | | | | | | | |

1. To reflect realistic conditions, some or all of the available resources may be estimated to be less than 100%.

**Table 4:** Mechanism 1, Data Set 1, T = 10, 40 available units

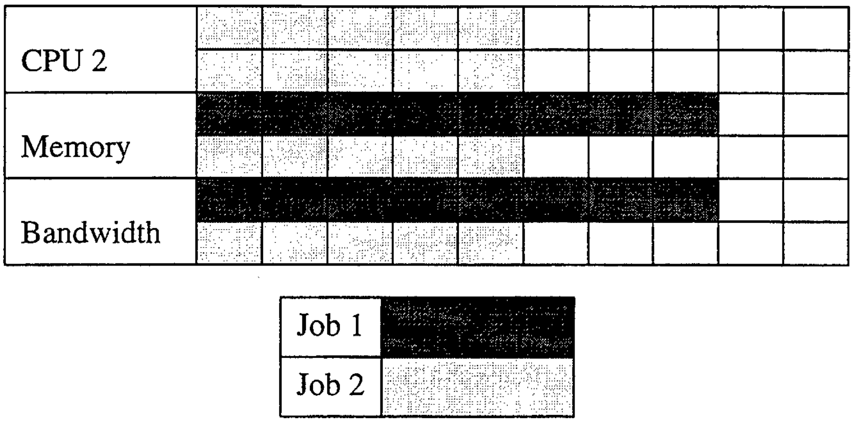
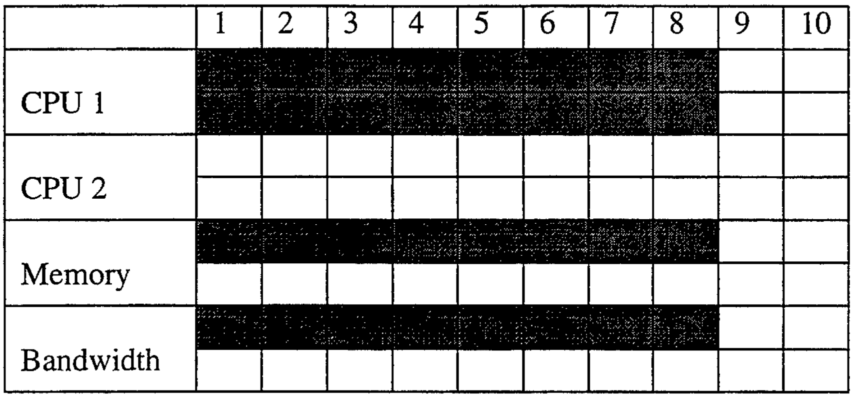



Table 5 shows the result of submitting data set 1 to mechanism 2. Here we see that the mechanism was not smart enough to utilize the second CPU, so Job 2 did not get scheduled at all. The efficiency of this mechanism is:

**Table 5:** Mechanism 2, Data Set 1, T = 10, 40 available units




*Efficiency* = 16/40 = .40

Table 6 and Table 7 show the result of submitting data set 2 to both mechanisms. Both jobs have enough time to finish the attempted jobs, so they both have the same efficiency. However, notice that in Table 6 , Mechanism 1 is not as efficient as it is in Table 4 , because the data set in Table 6 is easier.

*Efficiency* = 26/52 = .5

**Table 6:** Mechanism 1, Data Set 2, T = 13, 52 available units

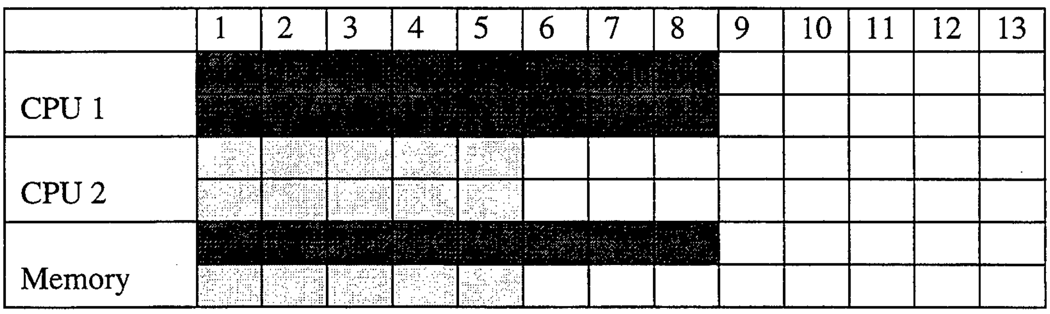


**Table 6:** Mechanism 1, Data Set 2, T = 13, 52 available units

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bandwidth | | | | | | | | | | | | | |

**Table 7:** Mechanism 2, Data Set 2, T = 13, 52 available units

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CPU 1 | | | | | | | | | | | | | |
| CPU 2 | | | | | | | | | | | | | |
| Memory | | | | | | | | | | | | | |
| Bandwidth | | | | | | | | | | | | | |

## 3 A Network Scheduler Benefit Function

In this section, we will pursue the notion of a network benefit function which reflects resource and priority policies and utilizes the efficiency framework already established. This serves as an example of our approach; other resource usage policies could also be characterized.

A simple objective function that measures how well a network resource scheduler performs from a QoS point of view can be expressed as follows [1] [5] , where *B* is an abstract per-job "user and system benefit function:"

$$max\sum_j B_j$$

That is to say, the network scheduler will be judged as to how well it meets the goals of the system and the users, as reflected in an expression of *B*.

### 3.1 Priority and Preference

Some jobs can produce output in different formats[1], where a given format (e.g., high resolution video) might be more resource consumptive than another format (e.g., low resolution video). A Quality of Service (QoS) scheduling mechanism might choose one format for a job over another, depending on varying network conditions (e.g., traffic congestion). The set of formats is represented by *F*. Different output formats may have different *preferences* (e.g., assigned by a user or "hard wired" as part of the application or job-scheduler database), and different levels of resource usage.

1. For the sake of simplicity, factors relating to "format" have not been included in the examples of the benefit function in Section 2. The inclusion of variable output formats for jobs results in an additional dimension of resource usage and efficiency. Development of this dimension in the expressions of $K_{ideal}$, $K_{actual}$, $E_{fficiency}$, and Table 1 , is left to the reader.

The formats for a job are assigned preferences ($p$) by the user

where $0 \le p \le 1$ and

$m_j$ is the number of <format, preference> pairs assigned for job $j$

$p_{fj}$ is the preference the user has assigned to format $f$, job $j$

the preferences for a job (j) add up to 1: $\forall j \in J \sum_{f=1}^{m_j} p_{fj} = 1$

Jobs are assigned priorities for use in resolving resource contention and allocation issues. For example, a critical production job might be assigned a higher priority than an optional job. Priorities are typically administratively assigned. In other words, priorities are used to order jobs, whereas preferences are used to order formats for a particular job.

$P_j$ is the priority of job $j$, where $0 \le P_j \le 1$

A network job scheduler should receive more credit in the benefit function for scheduling high priority and high preference jobs, as opposed to low priority or low preference jobs. We claim that a scheduler is intuitively doing a better job if important jobs, as judged by priority and preference, receive more attention (viz, resources) than unimportant jobs. How much weight the priorities and preferences are given is a matter of network scheduling policy.

## 3.2 Benefit Function

To begin with, the expression of *Efficiency* (see page 3) is simplified with a substitution:

let the expression of efficiency for a particular job ($j$) and format ($f$) be:

$$E_{fj} = \sum_{t=1}^{T} \sum_{r=1}^{[R]} Q_{fjrt}/([R]T)$$

then the expression of efficiency for a scheduler over all formats and jobs becomes:

$$Efficiency = \sum_{f=1}^{mj} \sum_{j=1}^{n} E_{fj}$$

We represent preference ($p$) and priority ($P$) in a benefit function by averaging them in with the expression of efficiency ($E$) as follows[1]:

1. Similar notions of priority, format and preference in measuring network efficiency have been proposed [6] for the MSHN project, however, "$B$" defines different relationships among these elements.

$$B = \frac{\sum_{j=1}^{n} \sum_{f=1}^{m_j} X_{fj}(P_j + p_{fj} + E_{fj})}{3n}$$

Where the characteristic function X is defined for f, j as:

$X_{fj} = 1$ if format $f$ was successfully delivered to job $j$ within time $T_j$, else 0

subject to: $\forall j \in J\left(\sum_{f=1}^{m_j} X_{fj} \leq 1\right)$

--at most one format is completed per job; f represents a particular <format, preference> pair[1]

## 3.3 Network Usage Policies

Network usage policies can vary widely. For example, restrictive usage policies (which include economic policies) attempt to moderate resource usage with usage-cost factors. In contrast, a priority-based policy emphasizes priority, diminishing the importance of a job's resource consumption. An ISP might manage its network of customers with an economic policy. A military information center might utilize a priority policy. ISP customers would be motivated not to "hog" network resources, by the cost incurred; whereas the military might want to ensure bandwidth to critical command-related jobs, at any cost.

Consider the analogy of water usage. The network is like a river, with citizens on the banks consuming water. The ISP and the military, in our example, are separate communities, and each have their own rivers. A set amount of water (viz., the resource) is flowing by and is available for use. The water that is not used flows past the community and is gone forever (here, our analogy breaks, as rivers are actually part of a globally replenishing cycle). So there is not a motivation to "conserve" water, other than to ensure that there is enough to go around at any given moment, and that it is allotted fairly. What is "fair" in a given community constitutes their water usage policy. Now, each community might have enough citizens to consume all of the water at periods of great usage. The ISP moderates the use of water, establishing a graduated policy: the first n units of water are charged at a low rate per gallon, while consumption above this limit is charged at a higher rate. The military (in this example) allows unlimited use to those with the highest priority, and divides up what is left among the lower-priority users.

If there is a drought, the river runs low. This is analogous to periods when there are equipment failures on a network.

## 3.4 Considering Resource Usage in the Benefit Function

In order to reflect a restrictive usage policy, we will modify the benefit function to give more credit to the scheduler for minimizing resource consumption. In other words, not only will the

1. This expression reflects the simplifying assumption that a job has the same <format, preference> pair throughout its execution; whereas an adaptive RMS might enable changing format during the execution of the job.

scheduler score high for maximizing priority, preference and scheduling goals, it will also do better if it meets these goals using fewer resources. This resource usage policy will facilitate the addition of new jobs to a set of running jobs, to the extent that it motivates the availability of resources (Venkatasubramanian and Nahrstedt consider both resource consumption and user satisfaction in developing their metric of video QoS [8] ).

A user may give a high preference to a job that requires high resource usage. We modify the preference term ($p$) with a representation of the job's ($j$) required resource usage[1], for a given format ($f$).

let $R_{fj} = 1 - K_{ideal.f.j}$

$p^*_{fj} = (p_{fj} + R_{fj}) / 2$

where $0 \le R_{fj} \le 1$, and a low $R_{fj}$ indicates high resource usage. Note that in our formulation we have given equal weight to $p_{fj}$ and $r_{fj}$; other weightings are possible

Table 8 shows an example of this modification to p. Here, we see preferences modified according to resource usage. There is no change for preferences whose values equal their resource usage values. A high preference job (.9) with high resource usage (.1) has a reduced preference; whereas, a low preference job (.1) with low usage (.9) results in an increased preference.

**Table 8: Preference Modified by Resource Usage**

| Resource Usage \ Job Preference | .1 | .3 | .6 | .9 |
|---|---|---|---|---|
| .1 | .1 | .2 | .35 | .5 |
| .3 | .2 | .3 | .45 | .6 |
| .6 | .35 | .45 | .6 | .75 |
| .9 | .5 | .6 | .75 | .9 |

Using this modified preference value, the scheduler will receive more credit for scheduling jobs that combine both high user preferences and low resource usages.

## 3.5 Final Expression of Benefit Function

Different priority policies can also be represented by giving more or less weight to the job's priority, with a policy weighting factor (integer W), $W \ge 0$ :

$$B = \frac{\sum_{j=1}^{n} \sum_{f=1}^{m_j} X_{fj}(WP_j + p^*_{fj} + E_{fj})}{n(W+2)}$$

1. $K_{ideal}$, from Section 2, is expanded here with respect to job and format

$0 \leq B \leq 1$

The benefit function now reflects the average of *Efficiency*, preference and priority of the jobs that are submitted, as modified by the policies for priority and resource usage.

## 4 Summary

An approach for characterizing resource usage has been presented. This approach was used to develop a metric for resource-usage efficiency. The metric is applicable in the context of our ongoing work to represent security in an RMS benefit function [6], and to articulate a costing framework for security, that, for example, might be provided to a resource management system [4]. We have illustrated an example of applying this metric to two simplified schedulers. The efficiency metric was then combined with expressions of priority and preference to create a benefit function ($B$) which would reflect scheduler effectiveness in meeting user and system goals. Resource usage was also used to modify the user preference variable to indicate an economic resource management model.

**Symbols Summary**

$B$ = benefit function

$C_{jrt}$ = amount of resource r required by job $j$ in time $t$

$E_{fj}$ = efficiency of a particular job and format

*Efficiency* = efficiency of a network job-scheduling mechanism

$F$ = {formats}

$J$ = {jobs}

$K_{ideal}$ = fraction of total resources required by a job data set

$K_{actual}$ = fraction of required resources scheduled for a job data set

$m_j$ = number of <format, preference> pairs for job $j$

$P_j$ = priority of job $j$

$p_{fj}$ = user preference for format $f$ in job $j$

$p^*_{fj}$ = user preference modified with respect to $R$

$Q_{jrt}$ = amount of resource r consumed by job $j$ in time $t$

$R_{fj}$ = inverse of $K_{ideal}$ for a given job/format

$R$ = set of schedulable resources

$[R]$ = number of distinct resources

$T$ = latest deadline in a job data set

$T_j$ = deadline for job $j$

$W$ = policy weighting factor for priorities

$X_{fj}$ = indicates if format $f$ was successfully delivered to job $j$ within time $T_j$

**References**


[1] Chatterjee, S., Sabata, B., Sydir, J. "ERDoS QOS Architecture," SRI Technical Report, ITAD-1667-TR-98-075, DARPA/ITO Contract Number N66001-97-C-8525, May 1998.

[2] Debra Hensgen, Taylor Kidd, David St. John, Matthew C. Schnaidt, H. J. Siegel, Tracy Braun, Jong-Kook Kim, Shoukat Ali, Cynthia Irvine, Tim Levin, Viktor Prasanna, Prashanth Bhat, Richard Freund, and Mike Gherrity, An Overview of the Management System for Heterogeneous Networks (MSHN), 8th Workshop on Heterogeneous Computing Systems (HCW '99), San Juan, Puerto Rico, Apr.1999

[3] Foster, I., and Kesselman, C., Globus: A Metacomputing Infrastructure Toolkit. Intl J. Supercomputer Applications, 11(2):115-128, 1997.

[4] Irvine, C., and Levin, T., Toward a Taxonomy and Costing Method for Security Metrics, NPS Technical Report, Forthcoming

[5] E. Jensen, C. Locke, and H. Tokuda, "A Time-driven Scheduling Model for Real-Time Operating Systems," in Proceedings of the IEEE Real-Time Systems Symposium, December 1985.

[6] Kim, Jong-Kook, Hensgen, D., Kidd, T., Siegel, H.J., St.John, D., Irvine, C., Levin, T., Prasanna, V., and Freund, R., Priorities, Deadlines, Versions, and Security in a Performance Measure Framework for Distributed Heterogeneous Networks, 8th IEEE International Symposium on High Performance Distributed Computing, Aug. 1999, Redondo Beach, Ca.

[7] Levin, T., and Irvine C., Quality of Security Service in a Resource Management System Benefit Function, NPS Technical Report, Forthcoming

[8] Vendatasubramanian, N. and Nahrstedt, K., "An Integrated Metric for Video QoS." ACM International Multimedia Conference, Seattle, Wa., Nov. 1997.

[9] Wright, R., Integrity Architecture and Security Services Demonstration for Management System for Heterogeneous Networks, Masters Thesis, Naval Postgraduate School, Monterey, CA, Sept. 1998.

[10] Wright, R., Shifflett, D., and Irvine, C. E., "Security Architecture for a Virtual Heterogeneous Machine," Proc. Computer Security Applications Conference, Dec. 1998.

## INITIAL DISTRIBUTION LIST

| | | |
|---|---|---|
| 1. | Defense Technical Information Center<br>8725 John J. Kingman Rd., STE 0944<br>Ft. Belvoir, VA 22060-6218 | 2 |
| 2. | Dudley Knox Library, Code 013<br>Naval Postgraduate School<br>Monterey, CA 93943-5100 | 2 |
| 3. | Research Office, Code 09<br>Naval Postgraduate School<br>Monterey, CA 93943-5138 | 1 |
| 4. | Defense Advanced Research Project Agency/<br>Information Technology Organization<br>3701 North Fairfax Drive<br>Arlington, VA 22203-1714 | 2 |
| 5. | Professor Cynthia E. Irvine<br>Code CS/IC<br>Department of Computer Science<br>Naval Postgraduate School<br>Monterey, CA 93943-5118 | 3 |